AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the above-identified application:

Claims 1-8 Canceled.

9. (Currently Amended): The system of claim [[8]] 23, wherein the SSL proxy includes a database operable to track information regarding a type of encryption scheme used to encrypt the ~~encrypted payloads~~ SSL packets.

10. (Canceled).

11. (Currently Amended): The system of claim [[8]] 23, wherein the SSL proxy tracks a message authentication code used to authenticate a message.

12. (Currently Amended): The system of claim [[8]] 23, wherein the SSL proxy is operable to encrypt packets sent from the server computer to the client computer.

13. (Currently Amended): The system of claim [[8]] 23, wherein a single end-to-end TCP connection exists between the client computer and the server computer and the source and destination address of the encrypted packets are unaltered.

14. (Canceled).

15. (Currently Amended) A method for processing SSL packets comprising:

initializing an SSL session between a client computer and a SSL proxy;

receiving a plurality of packets ~~including an encrypted portion~~ at the SSL proxy, each packet including at least a header and an encrypted portion;

determining if each of the received packets is a SSL packet by examining the header of each of the ~~second~~ received packets;

~~placing the SSL packet in a hold queue~~

decrypting SSL packets that are received in order;

placing SSL packets that are received out of order in a hold queue;

checking the hold queue to determine if the SSL packets placed therein are next in order for a given record;

releasing SSL packets from the hold queue if the SSL packets in the hold queue are next in order for a given record;

~~checking the hold queue to determine if all SSL packets expected for a given record have arrived;~~

decrypting the encrypted portion of each SSL packet released from the hold queue ~~once all the encrypted packets expected for the given record have arrived~~ to form decrypted SSL packets;

checking the decrypted SSL packets to determine if all SSL packets expected for a given record have arrived; and

outputting the decrypted packets to a server computer when all of the SSL packets expected for a given record have arrived.

16. (Currently Amended) The method of claim 15, wherein a message authentication code is checked to verify authenticity of the SSL packet set.

17. (Original) The method of claim 15, wherein non SSL packets are sent directly to the server.

18. (Canceled).

19. (Original) The method of claim 15, wherein the step of initializing further comprises initializing a single end-to-end TCP connection between the client computer and the server computer.

20. (Original) The method of claim 15, further comprising:

receiving packets with unencrypted data at a SSL proxy from the server computer;

encrypting the packets at the SSL proxy; and

sending the encrypted packets to the client computer.

21-22 (Canceled).

23. (New) A system for handling SSL traffic comprising:

a client computer running a web server operable to initiate an SSL session and to send data packets, each data packet including at least a header;

a server computer running a web browser operable to support communications with the client computer; and

a SSL proxy coupling the client computer and the server computer, the SSL proxy configured to receive the data packets sent from the client computer and operable, upon receipt therof, to:

(i) determine if each of the received packets is a SSL packet by examining the header of each of the received packets,

(ii) decrypt SSL packets that are received in order,

(iii) place SSL packets that are received out of order in a hold queue,

(iv) check the hold queue to determine if the SSL packets placed therein are next in order for a given record,

(v) release SSL packets from the hold queue if the SSL packets in the hold queue are next in order for a given record,

(vi) decrypt the encrypted portion of each SSL packet released from the hold queue to form decrypted SSL packets,

(vii) check the decrypted SSL packets to determine if all SSL packets expected for a given record have arrived, and

(viii) output the decrypted packets to a server computer when all of the SSL packets expected for a given record have arrived.